Shepherd, Ifan D. H. (2002) It's only a game: using interactive graphics middleware to visualise historical data. Society of Cartographers Bulletin, 36 (2) . pp. 51-55. ISSN 1469-5170 [Article]

First submitted uncorrected version (with author's formatting)

This version is available at: https://eprints.mdx.ac.uk/6586/

# It's Only a Game: Using interactive graphics middleware to visualise historical data

## Ifan D H Shepherd, Middlesex University Business School

**Abstract:**
This paper reports on the author's recent research into family-level mapping using data from manuscript notebooks compiled in the late nineteenth century by Charles Booth, author of the ground-breaking study of poverty in London: *Labour and Life of the People*. Using Booth's original data, the paper suggests ways in which recent visualisation technologies can be used to provide insights into historical data. In particular, it reports on the use of state-of-the-art videogame technology to develop an interactive 3-D visualisation engine to display point-located social data. The paper describes the software, its advantages and disadvantages for visualising historical data, and reports how it resolves a widespread and vexing problem in thematic mapping.

### Introduction
Digital mapping software is beginning to come of age. However, during the past decade, a variety of alternative technologies have emerged that offer effective visualisation of both 2-D and 3-D data, either in desktop environments or over the Internet. Alongside modern desktop mapping and GIS software are to be found an increasing range of software that can visualise spatial data, including: scientific data visualisation software; games engines; virtual reality software (including the now defunct Superscape product and its companion Web engine, Viscape); VRML (and GeoVRML) interpreters; and Java applets.

This paper reports on an experiment to create interactive 3-D visualisations of data for a late-Victorian survey of families in London. Rather than adopt conventional approaches rooted in geographic or cartographic technology, the project turned its back on existing approaches and instead developed software using a combination of the high-level programming language C and powerful videogames middleware (described below). The interactive performance of this software, even when visualising large datasets, is ensured by optimising the software for the modern, high-level 3-D graphics adapters available on current desktop and laptop PCs.

### Mapping and re-mapping Booth
The sample data used for this project come from a ground-breaking social survey carried out in the 1880s by Charles Booth (Booth 1889, 1889-1891). Booth's poverty maps of late-Victorian London are well known in the mapping community, and have achieved almost iconic status in the history of cartography (Hyde 1975, Topalov 1993). It is hardly surprising, then, that several attempts have been made over the past decade to re-map

Booth's original printed cartography: on paper (e.g. Reeder 1984), on CD-ROM (e.g. O'Day & Englander 1998) and over the Internet (e.g. Risotto 199?) The author's contribution to research in this area has been the creation of the first-ever family maps, based on data recorded in Booth's original notebooks (Shepherd 2000), and it is these data that provided the stimulus for the current visualisation experiments [1].

Data have been extracted from Booth's manuscript notebooks for a sample district located within the East End that includes well-known London landmarks such as Spitalfields Market and Middlesex Street (better known as Petticoat Lane). This district contains 79 streets, over 1700 dwellings and some 2400 families. The entire family data for these streets have been transcribed into a spreadsheet and imported in a geocoded form into a desktop GIS (MapInfo). (The geocoding was made possible by locating the street addresses recorded in Booth's notebooks for this area to precise locations on contemporary large-scale plans.) In order to a provide a suitable base map, the second-edition OS 25-inch base map of the area was scanned and stored as a raster image, and various boundary files were also created, including polygons for each of the coloured patches which were displayed on Booth's famous map to indicate the social condition of individual streets. Using this database, numerous family maps have been created (see Shepherd 2000 and 2003a for examples), largely using conventional point symbol mapping.

**Enter the dragon: an insidious mapping problem**

When the original family maps were created, using standard desktop mapping software (MapInfo), an unwelcome problem emerged: on all of the maps displayed, only a proportion of the point symbols representing individual families were visible. The cause of the problem was simple, though not immediately obvious: where several individuals, families or households lived at the same residential address, they shared identical geocodes. As a result, the point symbol maps created with standard mapping software concealed from view all but one of those individuals who live at the same address. (Because mapping software displays coincident point symbols in a 'stack', only the topmost symbol is visible to the map analyst. All others are concealed from view.)

This is an extremely thorny problem which does not admit of an entirely effective solution in the context of the 2-dimensional map, whether it be paper or digital. Feature coincidence is a potentially widespread problem in point symbol mapping, and gives rise to what the author has termed the 'hidden symbol' problem. A full review of this problem, its extent, causes and solutions, together with a discussion of the related problems of symbol overlap and symbol occlusion, is provided elsewhere (Shepherd 2003a).

The solution to the hidden symbol problem devised by the author involves a shift from 2-D mapping to 3-D visualisation. The use of 3-D symbolism and 3-D navigation within an interactive data visualisation environment has been demonstrated to provide a satisfactory resolution of the problem. In addition, the purpose-built interactive 3-D visualisation software provides a number of broader advantages for the visual exploration of historical data and, by extension, contemporary spatial data. This software is described in the following section.

**Using games technology to display point-located data**

Rather than continue with conventional mapping, GIS or scientific visualisation software, it was decided to develop real-time 3-D visualisation and navigation software based on videogame development technology. Most desktop mapping and GIS software is still designed for 2-D worlds and the static display of data, though extensions, add-ons and plug-ins are now available for most of the market leader programs to enable them to handle 3-D data and/or to provide some semblance of real-time interaction with spatial data through visualisation. The attractiveness of videogame technology lies in its ability to facilitate super-fast navigation through very large and often complex 3-dimensional worlds. Moreover, within the past few years, this technology has matured in ways that make it ever more capable of handling the types of spatial data (both vector and raster) that were once the exclusive preserve of digital mapping and GIS software. Increasingly, the spatial data handling capabilities of videogame engines are meeting and, in the 3-D arena, surpassing those of mainstream digital geotechnologies.

The obvious approach to harnessing this mature technology is to adapt an off-the-shelf games engine to handle real rather than imaginary world data. Games engines are becoming more and more sophisticated, and an increasing number of games (e.g. Grand Theft Auto 3 and The Getaway) involve detailed, large-scale representations of real (especially urban) landscapes. Some pseudo-realistic strategy games (e.g. SimCity) have been used for educational purposes, flight simulator games have been used for flight training of non-commercial pilots, and at least one games engine has been adapted for use as an educational tool (Foster 1997). It is possible to licence a games engine for commercial exploitation, or to reconfigure the spatial content of an existing game's database (e.g. Doom's WAD file) to handle real-world data.

For this research project, a different approach has been taken. Rather than use a standard games engine (the approach taken in the author's earlier research into the cognitive spatial strategies of games players), the technology underlying 3-D games engines has been used to write a specialist data visualisation program. In taking this decision, the author has stepped down a level in the games software hierarchy (which broadly consists of games engines, games development middleware, and low-level graphics libraries or APIs), and has selected one of the leading graphics middleware systems available for commercial games development: Renderware [2]. This system permits games software to be rapidly written in C or C++, which is the standard programming language used to create videogames with Renderware.

A major advantage of games middleware is that it combines a rich graphics library (i.e. a graphics API) with full access to a mainstream programming language (in the case of Renderware, this is C or C++). In addition, games middleware usually provides high-level graphics modelling and games-playing facilities. In the case of Renderware, these include: texture mapping (including complex surface shaders), polygon modelling, light mapping (including real-time dynamic lighting), LOD, object animation and 3-D sound. Renderware also enables applications to be written for a range of platforms, thus ensuring the widespread usability of resulting software. Currently, supported platforms include modern games consoles (including Sony's Playstation 2, Ninetendo's Gamecube, and Microsoft's Xbox) and PCs with suitable 3-D graphics adapters. The challenge for those wishing to develop visualisations using this approach is the expense of purchasing Renderware, and the learning curve involved in using C or C++ and the Renderware API. (It should be noted that the Renderware licence cost is typically a fraction of the cost of licensing a games engine, and programming skills in C or C++ are widely available among computer graduates.)

Using videogames middleware and a high-level programming language, it takes far less time and effort to write visualisation software than it did in the days of FORTRAN and lower-level graphics libraries. (Readers will doubtless have fond memories of writing applications software with such low-level graphics libraries as HPGL or Plot-10, higher-level graphics libraries such as GKS, PHIGS or PICASSO, or even dedicated mapping and spatial analysis libraries such as GAG.)

The Renderware system provides more than enough facilities for the development of interactive software to enable fly-abouts through 3-D space populated with 3-D point symbols displayed against a raster image base map. The completed Booth visualisation software includes functions to:

- import various data files from ASCII export files generated by the MapInfo and ArcView desktop mapping programs
- organise the imported data for speedy search and selection
- build and export efficient binary file versions of the various data components
- display 3-D symbols representing family data located on the base map image
- handle the mapping of multiple coincident families (using vertical displacement)
- display family attribute data thematically on the displaced 3-D point symbols
- facilitate 'fly-about' user navigation
- change display options and datasets during the course of the visualisation
- grab a screen dump at any point during the interactive map navigation and store it on file

These facilities were rapidly developed over the course of a few evenings and weekends [3].

**Using Renderware to visualise Booth data**

In order to import the MapInfo database into the Booth visualisation software, a number of modifications were made to the GIS files. The main focus was on pre-processing the large scanned base map image, which measured 2400 pixels horizontally by 3150 pixels vertically, to ensure effective screen rendering. First, the single GIF image was diced into standard square patches (or 'textures'), 256 pixels on each side, because standard-sized small texture tiles can be handled far more efficiently by modern graphics hardware. Textures also adapt more easily to LOD (level of detail) algorithms which will be needed in a future stage of the project when larger areas of London are to be visualised in real time. Next, the image was mip-mapped, in order to allow the image to be viewed from any distance without introducing aliasing artifacts, and bi-/tri-linear interpolation (a process similar to a 2-dimensional moving average filter) was applied to smooth the transitions between pixels. (The filter was clamped to all edges of each texture to avoid the visual artifacts caused by the default wraparound filter.) Because mip-mapping and linear interpolation operations are provided by most modern graphics adapters, their use considerably improves image quality during real-time fly-throughs with no appreciable performance penalty in terms of reduced frame rates.

The boundary and family point datasets were exported from MapInfo using the standard MID/MIF data formats, and the resulting text files were read by utility routines included in the visualisation software. Following initial tests, various modifications and improvements were made to the software. For example, a routine was added to display the boundaries of the street polygons in varied widths, with rounded inflexions, and another was written to reduce the greyscale intensity of the rasterised base map so that it did not visually overpower the point symbols used to represent the family locations.

**Discussion of Results**

The Booth visualisation software enables the effective interpretation of data displayed in three dimensions, because of its smooth 'fly-about' facilities. In order to interpret the spatial distribution of family characteristics, the navigator moves effortlessly around the 3-D map, using a number of navigation methods, until salient characteristics of the family distribution patterns have been identified and understood. The navigation is completely jitter-free, despite the detailed scanned image and the numerous point symbols displayed on screen. (For example, the map of cigar makers consists of 199 points, the map of family economic position consists of 1365 points, the map of family social class consists of 1478 points, and the map of all families includes 2128 points.) Thanks to the power of the graphics adapter, the software can run at high resolutions (the author's laptop runs the software at 1400 x 1050 pixels with 16-bit colour depth in full-screen mode and 32-bit colour depth in a window), without degrading the navigational experience.

**** Figure 1 about here ****

The visualisation software also provides a simple but highly effective 3-D solution to the seemingly intractable hidden symbol problem that frequently occurs in 2-D space. By plotting coincident families as vertical pillars of 3-D point symbols, the program enables all points in the database to be seen. In order to avoid point-of-view occlusion, the user has only to move around the scene to disambiguate the patterns on display. (See Sehpehrd 2003a for a detailed evaluation.)

The software offers various facilities to enable it to be scaled up to much larger databases. Tests with databases of varying sizes suggest that up to about 10,000 data points can be handled with the software running in its current mode of operation, in which every geometrical item appearing on screen is recreated afresh for each frame of the visualisation. Using a number of the technical 'tricks' used by games developers and available within the Renderware system (e.g. persistent mode rendering, LOD enabling, incremental data loading and unloading, world subdivision), it will be feasible to navigate in real time through databases which are one or two orders of magnitude larger than this. By using a simple project file mechanism, the user can switch rapidly from one database to another. (Project files are similar to the workspace mechanism in MapInfo, and permit the import of combinations of data files that make up the spatial database for a particular study area.)

## Conclusions

This paper has revealed how a commercial software development library, the Renderware product for videogames programming, has been used to visualise extensive spatial data in three dimensions. In addition to enabling easy navigation through 3-dimensional spaces in ways that permit effective interpretation of historical (and contemporary) spatial data, the software also provides a solution to the difficult hidden symbol problem frequently encountered in point symbol mapping. Further work is in progress to scale up the software to handle very large datasets, and to extend the functionality of the software to handle data exported from other desktop mapping and GIS programs. Experiments to date suggest that games development software provides a highly effective alternative to conventional GIS and mapping software for the interactive visualisation of large point feature datasets that need to be explored in 3-D space.


## Acknowledgements

I am delighted to acknowledge the help of my son, Iestyn Bleasdale-Shepherd, in building the Booth visualisation software. A senior software engineer at Criterion Software Ltd, he is a member of the Technical Team responsible for developing the core architecture of the Renderware software. I am also grateful to the organisers of the 2002 Annual SoC Technical Symposium for inviting me to present a paper entitled: '(Re-) Creating Charles Booth's Poverty Mapping of Late-Nineteenth Century London', from which the current paper was developed.


## Notes

1. Background information on Booth's maps and the family mapping project can be found on the author's Web site (http://mubs.mdx.ac.uk/~ Boothmaps/).

2. Renderware is a product of Criterion Software, a Canon company based in Guildford, UK. The software is used in a number of leading interactive videogames, including: Grand Theft Auto III and Vice City, Tony Hawk's Pro Skater 3, Pro Evolution Soccer, Airblade, and Burnout 1 and 2. It is also used in a number of more specialist titles such as: MX2002, City Crisis, Driven, and Super Bombad Racing.

3. Needless to say, this productivity owes a great deal to the experience of my expert programmer.

## References

Booth, C. (ed.) (1889) *Labour and Life of the People*, Volume 1, including map, London: Macmillan and Co.

Booth, C. (ed.) (1889-1891) *Labour and Life of the People,* 2 volumes, plus maps under separate cover, London: Macmillan and Co.

Foster, J. (1997) Doom and education, http://ci.cs.vt.edu/~mm/s97/projects/doom.txt (visited 17 November 2002). (The original report on the use of Doom in an information systems course in the US military, at: http://www.eecs.usma.edu/cs383/doom/default.htm, appears to have been withdrawn from this Web site.

Hyde, R. (1975) *Printed Maps of Victorian London 1851-1900* (Folkestone, Kent: Dawson)

O'Day, R. (1996-7) Charles Booth and social investigation in nineteenth century Britain, *Craft*, 16. Available at: http://www.arts.gla.ac.uk/www/ctich/Publications/craft16_1.htm (visited 17 November 2002).

Reeder, D. A. (1984) *Charles Booth's Descriptive Map of London Poverty 1889*, Publication 130, (London: Topographical Society).

Ahmad, S. (1999) Charles Booth's 1889 descriptive map of London poverty http://www.umich.edu/~risotto/home.html (visited 17 November 2002).

Shepherd, I. D. H. (2003a) Point Symbol Maps considered harmful: the Hidden Symbol Problem in Thematic Mapping and Spatial Data Visualization. Paper in preparation.

Shepherd, I. D. H. (2003b) Three Dimensional Visualisation for Personal Marketing and Geodemographic Data. Paper in preparation.

Shepherd, I. D. H. (2000) Mapping the poor in late-Victorian London: a multi-scale approach, in: J. Bradshaw and R. Sainsbury (eds) *Getting the Measure of Poverty: the early legacy of Seebohm Rowntree*, pp.148-176 (Aldershot: Ashgate)

Topalov, C. (1993) The city as *terra incognita*: Charles Booth's poverty survey and the people of London, 1886-1891, *Planning Perspectives*, 8(4), pp.395-425. (Originally published in *Geneses: Sciences Sociales et Histoire* (1991), 5, pp.5-34. English translation by A. Sutcliffe.)

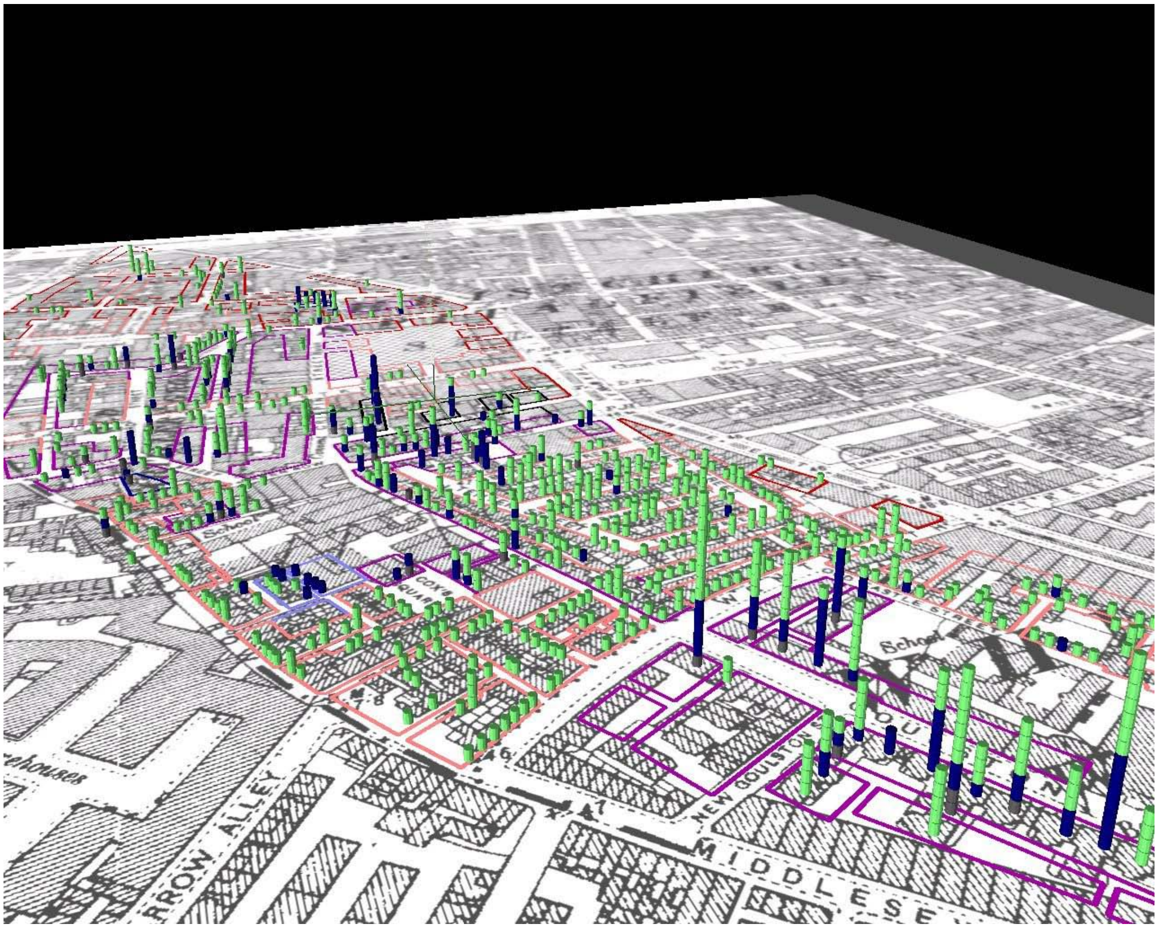


Figure 1. 3D display of nineteenth century geodemographic information for Whitechapel: (Source: Charles Booth notebooks, 1886-1888).